* For Examiners Reference

CLAIMS WITH INTERSTITIAL REFERENCES

FOR DISCUSSION PURPOSES ONLY, NOT TO BE INCLUDED IN FILE HISTORY

CLAIMS

What is claimed is:

(Claims 1-6 are Fig. 15a:)

1. A method of providing secure information, the method comprising regenerating a new encryption key **232** with an encryption key **224**, encrypted data **226**, and a hash vector based upon an encryption key **230**.

2. The method of claim 1 wherein the step of regenerating a new encryption key **232** with an encryption key **224**, encrypted data **226**, and a hash vector based upon an encryption key **230** comprises performing byte addition of an encryption key **224**, encrypted data **226**, and a hash vector based upon an encryption key **230**.

3. The method of claim 1 further comprising the step of hashing **228** a hash vector **226** based upon an encryption key.

4. The method of claim 3 wherein the step of hashing **228** a hash vector **226** based upon an encryption key comprises:

scanning indexed bytes of an encryption key; and

using indices and associated values of indices of an encryption key as indices of two bytes in a hash vector to be swapped **228**.

5. The method of claim 1 wherein the step of regenerating a new encryption key **232** with an encryption key **224**, encrypted data **226**, and a hash vector based upon an encryption key **230** comprises:

selecting a previously encrypted data record **226**; and

regenerating a new encryption key **232** with an encryption key **224**, selected encrypted data **226**, and a hash vector based upon an encryption key **230**.

6. The method of claim 5 wherein the step of selecting a previously encrypted data record comprises:

randomly selecting an index from the range [1, *t*-1] using a byte of an encryption key as a seed of random generation; and

selecting the previously encrypted data record **226** corresponding to the selected index.

(Claim 7 is Figs. 15a and 16:)

7. The method of claim 1 wherein the step of regenerating a new encryption key **232** with an encryption key **224**, encrypted data **226**, and a hash vector based upon an encryption key **230** comprises regenerating a new encryption key **314** with an encryption key **300**, previously encrypted data **302**, a hash vector based upon an encryption key **310**, and a received cipher **304**.

(Claims 8-9 are Fig. 15b:)

8. A method of providing secure information, the method comprising the steps of:

generating *n* encryption keys;

encrypting *n* tracks of data records with *n* tracks of parallel encryption; and

regenerating an encryption key with an encryption key, a hash vector based upon an encryption key, and selected encrypted data.

9. The method of claim 8 wherein the step of regenerating an encryption key with an encryption key, a hash vector based upon an encryption key, and selected encrypted data comprises:

randomly selecting an index from the range [1, *t*-1] using a byte of an encryption key as a seed of random generation; and

selecting the previously encrypted data record corresponding to the selected index.

(Claims 10-12 are Figs. 10 and 11:)

10. A method of providing secure information, the method comprising the steps of:

encrypting a data record with a hash vector based upon an encryption key **100**; and

regenerating an encryption key with an encryption key and encrypted data **102**.

11. The method of claim 10 wherein the step of encrypting a data record with a hash vector based upon an encryption key **100** comprises performing a logic operation on a data record **146** and a hash vector based upon an encryption key **148**.

12. The method of claim 11 wherein the step of performing a logic operation on a data record **146** and a hash vector based upon an encryption key **148** comprises performing an XOR operation on a data record **146** and a hash vector based upon an encryption key **148**.

(Claims 13 – 14 are Fig. 13:)

13. The method of claim 10 further comprising the step of decrypting encrypted data, comprising performing a logic operation on an encrypted data record **164** and a hash vector based upon an encryption key **166**.

14. The method of claim 13 wherein the step of performing a logic operation on an encrypted data record **164** and a hash vector based upon an encryption key **166** comprises performing an XOR operation on an encrypted data record **164** and a hash vector based upon an encryption key **166**.

(Apparatus:)

15. A system for providing secure information, the system comprising:

a source node U_s;

a destination node U_d;

a data stream created at said source node;

means for encrypting data of said data stream with a hash vector based upon an encryption key **148 (see Fig. 11)**; and

means for regenerating a new encryption key **232** with an encryption key **224**, encrypted data **226**, and a hash vector based upon an encryption key **230 (see Fig. 15a)**.

(Claims 16 – 20 are Figs. 2, 3a, 3b and 4:)

16. A method of authenticating one system node to another system node, the method comprising the steps of:

generating an authentication key ***DAK*** at a node ***CA*, 12**;

transmitting the authentication key to another node U_s ***or*** U_d, **12**; and

starting a daemon at each node ***CA and U*** for regenerating a new authentication key **222** with an authentication key **216**, an auxiliary key **218**, and a hash vector based upon an authentication key **220**, and maintaining a corresponding number-regeneration-counter at each node **14**, **15**.

17. The method of claim 16 wherein the step of regenerating a new authentication key **222** with an authentication key **216**, an auxiliary key **218**, and a hash vector based upon an authentication key **220** comprises performing byte addition of an authentication key **216**, an auxiliary key **218**, and a hash vector based upon an authentication key **220**.

18. The method of claim 16 further comprising the step of generating an auxiliary key ***K*, 200 or 210** from at least one key selected from the group consisting of encryption keys **204**, authentication keys **202**, **212**, **214**, and a hash vector based upon an authentication key.

19. The method of claim 18 wherein the step of generating an auxiliary key ***K*, 200 or 210** comprises generating an auxiliary key **200** by performing byte addition of an authentication key **202**, an encryption key **204**, and a hash vector based upon an authentication key **206**.

20. The method of claim 18 wherein the step of generating an auxiliary key ***K*, 200 or 210** comprises generating an auxiliary key **210** by performing byte addition of two or more authentication keys **212**, **214** and a hash vector based upon an authentication key.

(Claim 21 is Fig. 17a:)

21. A method of validating data integrity, the method comprising the steps of:

buffering an encryption key and a hash vector based upon an encryption key at a source node **316**;

encrypting a data record using a hash vector based upon an encryption key of a first point in time to yield a cipher record at a source node **318**;

transmitting the encrypted data record to a destination node **318**;

receiving a cipher from a destination node **320**;

decrypting the received cipher from the destination node with a hash vector based upon an encryption key of a second point in time **322**; and

comparing the decrypted received cipher to a data record **324**.

(Claim 22 is Fig. 17b:)

22. The method of claim 21 further comprising the steps of:

buffering an encryption key and a hash vector based upon an encryption key at a destination node **330**;

encrypting a data record using a hash vector based upon an encryption key of a second point in time to yield a cipher record at a destination node **332**;

transmitting the encrypted data record to a source node **332**;

receiving a cipher from a source node **334**;

decrypting the received cipher from the source node with a hash vector based upon an encryption key of a first point in time **336**; and

comparing the decrypted received cipher to a data record **338**.

(Claims 23- 29 are Figs. 6, 7 and 8:)

23. A method of synchronizing one node to another node, the method comprising the steps of:

receiving a request from a first user U_s to communicate with a second user U_d along with an authentication key number regeneration count and a hashed value of an authentication key number regeneration count **16**;

requesting an authentication key number regeneration count and a hashed value of an authentication key number regeneration count from a second user U_d **18**;

comparing a central authority authentication key number regeneration count to a user authentication key number regeneration count **22 and 36, 38**; and

aligning the authentication keys of a user and a central authority node according to the comparison **42, 44 and 46**.

24. The method of claim 23 wherein the step of receiving a request from a first user U_s to communicate with a second user U_d along with an authentication key number regeneration count and a hashed value of an authentication key number regeneration count **16** comprises receiving a request from a first user U_s to communicate with a second user U_d along with an authentication key number regeneration count and a hashed value of an authentication key number regeneration count **16** encrypted with a static key K **16**.

25. The method of claim 23 wherein the step of requesting an authentication key number regeneration count and a hashed value of an authentication key number regeneration count from a second user U_d **18** comprises requesting an authentication key number regeneration count and a hashed value of an authentication key number regeneration count encrypted with a static key K from a second user U_d **18**.

26. The method of claim 23 further comprising the step of authenticating the identity of the first and second user. **(Figs. 8a and 8b.)**

27. The method of claim 26 wherein the step of authenticating the identity of the first and second user comprises:

generating a nonce ***N*** at a central authority node **50**;

encrypting a nonce with a hash vector of an authentication key **50**;

transmitting an encrypted nonce to a user node **50**;

decrypting an encrypted nonce at a user node **64**; and

comparing a decrypted nonce with a nonce **66**.

28. The method of claim 27 wherein the step of encrypting a nonce with a hash vector of an authentication key **50** comprises:

generating additional authentication keys **50**; and

encrypting a nonce with a hash vector of an additional authentication key **50**.

29. The method of claim 27 further comprising the steps of:

generating additional authentication keys;

transmitting a nonce encrypted with a hash vector of an additional authentication key to a central authority **68**;

decrypting an encrypted nonce at a central authority **54**; and

comparing a decrypted nonce with a nonce at a central authority **56**.